SCOPE OF VARIABLES:

Scope of Variables in Java:

In programming, scope of variable defines how a specific variable is accessible within the program or across classes. In this section, we will discuss the scope of variables in Java.

Scope of a Variable:

In programming, a variable can be declared and defined inside a class, method, or block. It defines the scope of the variable i.e. the visibility or accessibility of a variable. Variable declared inside a block or method are not visible to outside. If we try to do so, we will get a compilation error. Note that the scope of a variable can be nested.

- We can declare variables anywhere in the program but it has limited scope.
- A variable can be a parameter of a method or constructor.
- A variable can be defined and declared inside the body of a method and constructor.
- It can also be defined inside blocks and loops.
- Variable declared inside main() function cannot be accessed outside the main() function

Types of Variables:

There are three types of variables in Java:

- local variable
- instance variable
- static variable

1) Local Variable:

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

A local variable cannot be defined with "static" keyword.

2) Instance Variable:

A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static.

It is called an instance variable because its value is instance-specific and is not shared among instances.

3) Static variable:

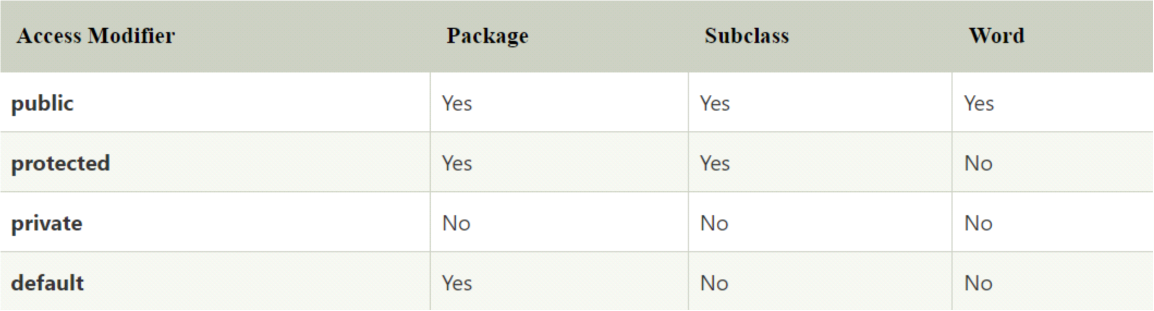
A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.

Here is Types of Variables based on there Scope:

- **Member Variables (Class Level Scope)**
- **Local Variables (Method Level Scope)**

1.>Member Variables (Class Level Scope)

These are the variables that are declared inside the class but outside any function have class-level scope. We can access these variables anywhere inside the class. Note that the access specifier of a member variable does not affect the scope within the class. Java allows us to access member variables outside the class with the following rules:



Access Modifier	Package	Subclass	Word
public	Yes	Yes	Yes
protected	Yes	Yes	No
private	No	No	No
default	Yes	No	No

Syntax;

```
public class DemoClass
{
//variables declared inside the class have class level scope
int age;
private String name;
void displayName()
{
//statements
}
int dispalyAge()
{
//statements
}
char c;
}
```

2.>Local Variables (Method Level Scope)

These are the variables that are declared inside a method, constructor, or block have a method-level or block-level scope and cannot be accessed outside in which it is defined. Variables declared inside a pair of curly braces {} have block-level scope.

Declaring a Variable Inside a Method

```
public class DemoClass1
{
void show()
{
//variable declared inside a method has method level scope
int x=10;
System.out.println("The value of x is: "+x);
}
public static void main(String args[])
{
DemoClass1 dc = new DemoClass1();
dc.show();
}
}
```

MCQ QUESTIONS:

1. Where to declare local variables ?

A. It declares inside the class and outside of method.

B. It declares inside the class with static prefix and outside of method.

C. It declares inside the method.

Answer: C

2. Where to declare instance variables ?

A. It declares inside the class and outside of method.

B. It declares inside the class with static prefix and outside of method.

C. It declares inside the method.

Answer: A

3. Where to declare class variables ?

A. It declares inside the class and outside of method.

B. It declares inside the class with static prefix and outside of method.

C. It declares inside the method.

Answer: B

4. Which of following can not be used as variable name in Java programming language ?

A. literal

B. Keywords

C. Identifier

D. Identifier and Keywords

Answer: B

5. A name of variable is known as

A. Data Type

B. Constant

C. Identifier

Answer: C

6. Which is the invalid variable name ?

A. tenthTeam

B. TenthTeam

C. 10thTeam

D. None of Above

Answer: C

7. A variable name should not begin with

A. A number

B. Special characters but underscore and dollar sign

C. All of above

Answer: C

8. Which is the valid variable name ?

A. 5thTeam

B. &5thteam

C. _5thTeam

D. None of Above

Answer: C

9. Which are the invalid variable names ?

A. 1stLevel

B. first Level

C. *firstLevel

D. All of Above

Answer: D

10. Which are the valid variable names ?

A. $1stLevel

B. _1stLevel

C. £1stLevel

D. All of Above

Answer: D

11. A variable declared inside the method is known as.

```
public class TestMain {
   public static void main(String[] args) {
      String  msg = "Hello World"; // msg variable name
      System.out.println(msg);
   }
}
```

A. Instance Variable

B. Class Variable

C. Local Variable

Answer: C

12. A variable declared outside of a method is known as.

```
public class TestMain {
   String  msg = "Hello World"; // msg variable name
   public static void main(String[] args) {
      TestMain test = new TestMain();
      System.out.println(test.msg);
   }
}
```

A. Instance Variable

B. Class Variable

C. Local Variable

Answer: A

13. A variable declared outside of a method with prefix static keyword is known as

```
public class TestMain {
   static String  msg = "Hello World"; // msg variable name
   public static void main(String[] args) {
      System.out.println(msg);
   }
}
```

A. Instance Variable

B. Class Variable

C. Local Variable

Answer: B

14. A variable's _____________ indicates where the variable can be used in an application code.

a) Scope

b) Lifetime

c) Place

d) Position

Answer: a

15. A variable's _______________ indicates how long the variable stays in the computer's internal memory.

a) Scope

b) Lifetime

c) Place

d) Position

Answer: b

16. Variables declared in a form's declaration section have _____________ scope.

a) Class

b) Procedure

c) Block

d) Object

Answer: a

17. Variable declared in a procedure have ______________ scope.

a) Class

b) Procedure

c) Block

d) Object

Answer: b

18. Variables declared in a block have _____________ scope.

a) Class

b) Procedure

c) Block

d) Object

Answer: c

19. When you declare a variable in the form’s declaration section, it is called as ______________

a) Class-level variable

b) Object variable

c) Procedure-level variable

d) Auto variable

Answer: a

20. Lifetime of a class-level variable is till ________________

a) A procedure ends

b) An application ends

c) A block ends

d) Within the control section

Answer: b

21. A class-level variable is declared using ______________ keyword.

a) Private

b) Public

c) Dim

d) Static

Answer: a

22. A procedure-level variable is declared using ____________ keyword.

a) Private

b) Public

c) Dim

d) Static

Answer: c

23. __________ is a variable that remains in memory, and retains its value even after the procedure declaring it ends.

a) Procedure-level variable

b) Class-level variable

c) Static variable

d) Dynamic variable

Answer: c

24. Static variable can only be used in the __________ in which it is declared.

a) Class

b) Any procedures in the form

c) Any form

d) Only the Procedure it is declared in

Answer: d

25. A __________ cannot change its value while the application is running.

a) Constant variable

b) Class variable

c) Named constant

d) Static constant

Answer: c

26. Named constants are used because ___________

a) Because they take up less space in memory

b) Because they make documentation and code modification easier

c) Because they can be easily accessed by procedures

d) Because they are easily declared

Answer: b

27. What is wrong with the expression “const dblPi As Double = 3.141593”?

a) Double is written

b) Name should be DBLPi

c) “const” starts with a “C”

d) Value is too large

Answer: c

28. What is wrong with the expression “static dblPi AsDouble ”?

a) Double is written

b) Nothing is wrong

c) “Static” starts with a “S”

d) Value is not given

Answer: c

29. What happens when a procedure containing the following statements ends?

```
Dim decSales As Decimal=12.2

Static decTotal As Decimal=13.5
```

a) Both values lose their values

b) Dim variable loses its value

c) Static variable loses its value

d) Both variables retain their values

Answer: b

30. To declare a class-level named constant you declare it as __________

a) Public

b) Dynamic

c) Private

d) Static

Answer: c